

IN THE CLAIMSListing of Claims:

1. (currently amended) A method for managing a link stack comprising the steps of:
 2. ~~for a first type instruction~~:
3. setting a first data value, corresponding to a portion of a first address stored in
4. said link stack, [[in]] into a first portion of an entry in a queue having a plurality of
5. entries in response to a first type operation; [[and]]
6. ~~for a second type instruction~~:
7. setting a current value of a first pointer into of said link stack [[in]] into a first
8. register in response to a second type operation; and
9. setting said current value of said first pointer [[in]]into a second portion of said
10. entry in said queue.
 1. 2. (currently amended) The method of claim 1 further comprising the steps of:
 2. ~~for said second type instruction~~, reading from said link stack a second address
3. stored at a stack entry at said current value of said first pointer in response to said second
4. type operation; and
5. storing said second address in a second register.
 1. 3. (currently amended) The method of claim 1 wherein said first type instruction
2. operation is a “push” type instruction operation [[an]] and said second type instruction
3. operation is a “pop” type instruction operation.
 1. 4. (currently amended) The method of claim [[1]] 2 further comprising the step of, ~~for a~~
2. ~~first type operation~~, setting said first data value in a third register for said first type
3. operation.
 1. 5. (currently amended) The method of claim 4 further comprising the step of, ~~for third~~
2. ~~type instruction~~, setting a second data value from said second register in a said first
3. portion of said entry in said queue in response to a third type operation.

1 6. (currently amended) The method of claim 1 wherein said steps recited therein are
2 performed in response to a fetch of an instruction using a corresponding one of said first
3 type instruction operation and said second type instruction operation.

1 7. (currently amended) The method of claim 2 further comprising the steps of:
2 receiving a pointer value in response to a pipeline flush operation, said pointer
3 value operable for pointing into said queue; and
4 setting a current pointer value of said first pointer to a value in said second
5 portion of an entry in the queue pointed to by said pointer value.

1 8. (currently amended) The method of claim 7 further comprising the steps of:
2 retrieving a value from said second register; and
3 comparing said value from said retrieving step with a value from said second
4 portion of said entry in the queue pointed to by said pointer value.

1 9. (currently amended) The method of claim 8 further comprising the steps of:
2 in response to a compare match in said comparing step, comparing said current
3 value of said first pointer with a data value in said first register; and
4 in response to a compare match of said current value of said first pointer and said
5 data value in said first register, setting said data value in said second register [[in]] into
6 said link stack at a location pointed to by said current value of said first pointer
7 decremented by one.

1 Claims 10-17 (canceled)

1 18. (currently amended) A data processing system comprising:
2 a central processing unit (CPU), said CPU including:
3 a link stack; and
4 first logic operable for ~~, for a first type instruction~~, setting a first data value
5 corresponding to a portion of a first address stored in said link stack [[in]] into a first

6 portion of an entry in a queue having a plurality of entries in response to a first type
7 operation, and for a second type instruction, setting a current value of a first pointer into
8 said link stack in a first register in response to a second type operation, and setting said
9 current value of said first pointer in a second portion of said entry in said queue.

1 19 (currently amended) The system of claim 18 wherein said CPU further comprises,
2 second logic operable for, for said second type instruction operation, reading from said
3 link stack a second address stored at a stack entry at said current value of said first
4 pointer, and storing said second address in a second register.

1 20. (currently amended) The system of claim 18 wherein said first type instruction is a
2 “push” type instruction operation [[an]] and said second type instruction operation is a
3 “pop” type instruction operation.

1 21. (currently amended) The system of claim [[18]] 19 wherein said CPU further
2 comprises, third logic operable for, for a first type operation, setting said first data value
3 in a third register for a first type operation.

1 22. (currently amended) The system of claim 21 wherein said CPU further comprises
2 fourth logic operable for, for third type instruction operation, setting a second data value
3 from said second register in a said first portion of said entry.

1 23. (currently amended) The system of claim 18 wherein said first logic sets said first
2 data value, for said first type instruction, and sets said current value of said pointer, for
3 said second type instruction operation in response to a fetch of a corresponding one of
4 said first type instruction operation and said second type instruction operation.

1 24. (currently amended) The system of claim 19 wherein said CPU further comprises:
2 fifth logic operable for receiving a pointer value in response to a pipeline flush
3 operation, said pointer value operable for pointing into said queue; and

4 sixth logic operable for setting a current pointer value of said first pointer to a
5 value in said second portion of an entry in the queue pointed to by said pointer value.

1 25. (currently amended) The system of claim 24 wherein said CPU further comprises:
2 seventh logic operable for retrieving a data value from said second register; and
3 eighth logic operable for comparing said value from said retrieving step with a
4 data value from said second portion of said entry in the queue pointed to by said pointer
5 value.

1 26. (currently amended) The system of claim 25 wherein said CPU further comprises:
2 ninth logic operable for, in response to a compare match in said comparing step,
3 comparing said current value of said first pointer with a data value in said first register;
4 and

5 tenth logic operable for, in response to a compare match of said current value of
6 said first pointer and said data value in said first register, setting said data value in said
7 second register [[in]]into said link stack at a location pointed to by said current value of
8 said first pointer decremented by one.

1 27. (currently amended) The system of claim 18 further comprising system memory
2 coupled to said CPU, said system memory operable for storing a program of instructions
3 including instructions of said first type operations and said second type operations.

1 Claims 28-48 (canceled)